AD-A227 799

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

90 10 12 088

AU-AFIT/LS/TR-90-1

AU-AFIT/LS/TR-90-1

AIR FORCE INSTITUTE OF TECHNOLOGY

AN INTRODUCTION TO EXPERT SYSTEMS AND KNOWLEDGE ACQUISITION TECHNIQUES

James R. Heatherton, Captain, USAF
Todd T. Vikan, Captain, USAF

September 1990

Approved for public release; distribution unlimited

The opinions and conclusions in this paper are those of the author and are not intended to represent the official position of the DOD, USAF, or any other government agency.

<u>Preface</u>

This report is the by-product of information collected by the authors during research into expert system technology conducted at the Air Force Institute of Technology. That research involved methods for selecting appropriate tools (or "knowledge acquisition techniques") to collect information from experts. In the course of the research, ~~we~~ discovered that no single publication discussed all of the collection techniques that a knowledge engineer might want to evaluate.

This brief report attempts to remedy that deficiency by consolidating into one document the primary knowledge acquisition techniques used today. For each technique, ~~we~~ have provided a short description, evaluation, and bibliography for individuals who want to evaluate a technique in greater depth. The discussion of techniques is introduced by an overview of some issues and architectures of expert system design.

We hope that this survey will be useful to anyone starting to work with expert systems, as well as to busy managers who want to be certain they have selected the best tool for the important job of knowledge extraction.

Capt James R. Heatherton

Capt Todd T. Vikan

Table of Contents

List of Figures

List of Tables

KNOWLEDGE ACQUISITION FOR EXPERT SYSTEMS

I. Introduction

General Issue

Advances in computer software technology, coupled with the declining price of computer hardware, have put powerful computer capabilities in the hands of even the smallest of organizations. One application for this computer capability is the expert system. An expert system is a computer program that attempts to mimic an expert's decision processes to provide solutions to specific problems (34:736).

To the uninitiated, the design, development, and implementation of expert system technology may seem like a formidable task. While there are many difficulties to overcome in constructing a usable expert system, many of the current expert system development tools put the capability within reach of virtually any competent computer user. Not only are the latest tools easier to use, but the cost continues to decrease. A key step in developing an expert system is the acquisition of expert knowledge to construct the system's knowledge base. The acquisition of expert knowledge is perhaps the greatest hurdle facing the knowledge engineer. Even though many knowledge acquisition techniques exist, little empirical evidence is available to indicate which knowledge acquisition techniques are better for specific types of knowledge.

The purpose of this paper is to review the literature written about expert systems and knowledge acquisition, and to emphasize the importance of knowledge acquisition as a process in the development of expert systems. It is intended to reduce some of the mystique that surrounds knowledge acquisition, to consolidate knowledge acquisition information, and to identify practical knowledge acquisition methods.

Justification for the Search and Review

The greatest single application of artificial intelligence technology today is in expert systems. More time and money is being invested in expert systems than in any other segment of the artificial intelligence business (35:16). One of the most important and difficult activities in the development of these expert systems is the process of knowledge acquisition (22:269; 29:152; 3:144). Much has been written about how to build expert systems, but little has been written about knowledge acquisition (29:152).

Method of Treatment and Organization

To establish the importance of knowledge acquisition in the development process of expert systems, the first two sections of this review will describe expert systems and their applications. The first section defines an expert system and describes its major components. The second section presents some criteria to help determine the applicability of expert systems and provides examples of expert system applications. The third major section deals with the steps in the development of expert systems. It presents several frameworks for expert system development, identifies some of the current expert system programming tools, expert selection and, most importantly, knowledge acquisition. The final section of this review describes some of the problems the knowledge engineer may face when acquiring expert knowledge.

II. Expert System Description

General Definition and Description

Expert systems are computer programs designed to simulate the cognitive problem-solving behavior of human experts in a specified, well-defined area. These programs contain stores of knowledge, usually in the form of facts and rules, together with procedures for processing this knowledge to infer solutions to problems normally requiring the attention of a human expert. Additionally, the user is able to solicit information pertaining to the problem at hand and to obtain explanations about the way the program behaves (38:4-5). Expert systems vary considerably from conventional computing systems. Conventional computer programs are based on algorithmic, clearly defined, step-by-step procedures for solving problems. Expert systems are able to reason about data and draw conclusions by employing heuristic rules. Heuristics are sometimes characterized as the "rules of thumb" that one acquires through practical experience to solve everyday problems (35:17). For example, a fisherman learns from experience that a flock of diving sea gulls combined with the smell of fish oil indicates the strong likelihood that a school of fish is feeding just below the sea gulls. The expert fisherman will abandon random searching for fish and move to the area near the sea gulls (38:5).

Components of an Expert System

There are three primary integrated components in an expert system. These components are a knowledge base to store information, an inference engine to draw conclusions from the information, and a user interface to gather and disseminate the information. Figure 1 illustrates the relationship between these components and the user they serve.

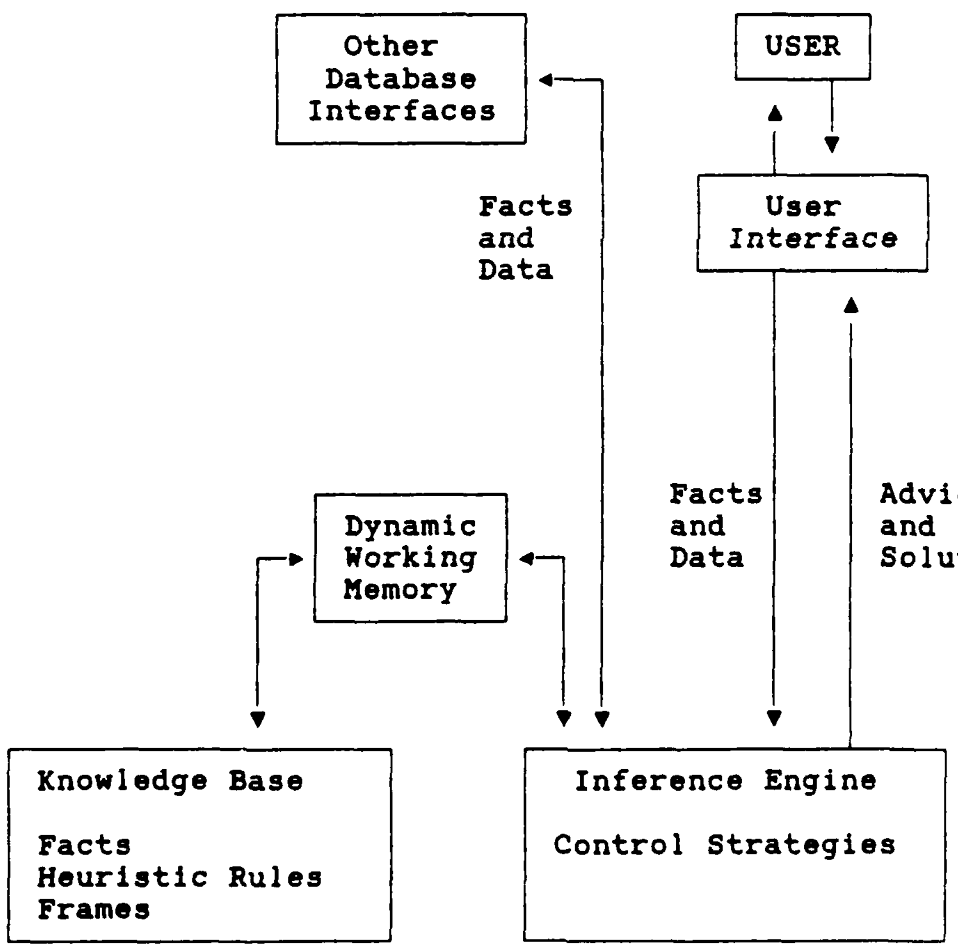

Figure 1. Components of an Expert System. (Adapted from 2:27)

The Knowledge Base. The heart of any expert system is its knowledge base. The knowledge base contains the facts, rules, and other knowledge required to solve a problem (40:18). There are several methods for representing knowledge in the knowledge base. Among these are semantic networks, frames, and productions rules (38:8-9). Semantic networks are graphical depictions of knowledge that show hierarchical relationships between objects (38:10-12). Frames are similar to semantic networks, but they contain a larger, more detailed block of knowledge about a particular object (38:10-12). This detailed data is given in a sub-element called a slot (38:10-12). Production rules are two-part statements that embody small pieces of knowledge. The first part of the rule expresses a situation or premise while the second part states a particular action or conclusion that applies if the situation or premise is true (38:13-15). Most commercial and experimental expert systems use an IF-THEN production rule format such as: IF it is raining outside; THEN take an umbrella with you to work (38:15).

The Inference Engine. The inference engine implements the search and pattern matching strategy of the expert system (40:22-23). It is sometimes called a rule interpreter because its operation is somewhat like a software interpreter for a computer language (38:15-16). The two inference strategies commonly used in expert system inference engines are forward chaining and backward chaining (38:15).

Forward Chaining. In forward chaining, the inference engine begins with what is known about current conditions and works forward in an attempt to infer inductively what is unknown (38:15). Forward chaining systems are said to be data-driven and reason in a bottom-up or inductive fashion (38:15). This kind of system is most useful in problem domains where there are many possible goals and all that is known to the program are details of current conditions (38:14-15).

Backward Chaining. Backward chaining systems operate in the opposite fashion. According to Teft, these inference engines start by looking at a fact in the form of a hypothesis. The inference engine then seeks to find evidence to support one or more of these hypothesis. This type of inference engine works top-down and is driven by the hypothesis. These system are most useful when a relatively small number of goals are considered to be strong candidates as solutions to the problem. The backward chaining system attempts to prove these goals by finding evidence for their conditions (38:17).

An Example. In his book Programming in Turbo Prolog (38:17), Lee Teft uses the concept of a consultation between a doctor and patient to illustrate the difference between forward and backward chaining strategies. If the patient were to describe vague symptoms of not feeling well, and there were no obvious outward signs to indicate the nature of the illness, the doctor might "chain forward" by asking a series of questions and performing a series of tests. Depending on the answers to the questions and the results of the tests, the doctor could then inductively arrive at a diagnosis from the facts that he gathered. On the other hand, if the patient was to display highly visible symptoms which might lead the doctor to a good first diagnosis, the doctor might then elect to "chain backwards" asking a series of questions or by conducting tests to support his initial diagnosis (38:17).

User Interface. The final element of the expert system is the user interface. According to Waterman, it is the part of the computer program that allows the user to communicate with the system. The user interface asks questions or presents menu driven choices for entering initial information. It provides a means of communicating the answer or solution once it has been found. Most expert systems also provide the user with a summary of the steps used in arriving at the solution. This allows the user to follow the logic involved and to become more comfortable with the outcome (40:18).

III. Expert System Application

Criteria

An expert system is problem solving software that contains the knowledge of one or more experts from a specific domain. While expert system technology has proven to be useful for many diverse tasks, there are recognized criteria and limits to the application of this technology. These criteria and limits are discussed in the following section.

While the acquisition of expert knowledge to put into an expert system knowledge base may be the most difficult step in the development of an expert system, identifying and selecting an appropriate general problem area (also known as domain) and specific task may be the most critical steps (30:26). Some of the existing literature emphasizes the characteristics of the knowledge that the expert system needs, other literature emphasizes the nature of the task, and still other literature combines features of both to develop a measure by which a domain and task can be evaluated to determine their suitability for expert system technology.

There are a number of desirable features that provide general guidelines to problem selection. Expert system technology is best applied to well-defined problems, but the decision processes within that problem area should be semi-structured or unstructured (10:1). These decision processes are those for which there is no programed or documented response. According to Chignell, expert systems are most appropriate when "rule-based task analyses of the system of interest are available or easily derived" (5:391). Expert systems should be restricted to narrowly defined processes, these processes should not be overly difficult, and they should not rely on a great deal of "common sense" (5:384).

Keim and Jacobs. Keim and Jacobs have developed a list of undesirable features of a problem domain and task that may render expert system technology unsuitable.

1. Decisions involving too few rules. (fewer than 10)
2. Decisions involving too many rules. (more than 10,000)
3. Well structured decisions.
4. Decisions solved by human abilities such as pattern recognition.
5. Broad, superficial knowledge domain decisions.
6. Decisions involving the latest technology.
7. Areas of controversy among domain experts. (21:5-6)

Keim explains that decisions involving fewer than 10 rules are much too simple to warrant financial and manpower investment in an expert system. Decisions involving more than 10,000 rules may be too complicated and would take too long to develop. Keim supports Eide's statement that expert system technology is better suited to semi-structured and unstructured or unregulated decision processes than structured processes. Decisions involving human perceptual abilities are currently beyond applied expert system technology. Problem domains involving the latest in technology are going to have few if any "experts" from which to acquire knowledge. Lastly, areas involving conflict among existing experts may cause programming and verification difficulties as conflicting rules are encoded into the knowledge base (21:12-13).

Waterman. Waterman developed a series of questions one can use to evaluate a problem domain and task to determine their suitability for expert system technology. These questions are arranged into three groups that help determine the possibility of project success, identify a sound justification, and determine the appropriateness of the effort (40:127).

Possible. Waterman suggests seven criteria that must all be satisfied if the project is to be possible (Figure 2). The most important criterion, however, is establishing that an expert does exist and is willing to participate in the project. Waterman feels that without a true expert's knowledge from which to develop the knowledge base, the possibility of project success is greatly reduced (40:128).

Justified. Waterman identifies five reasons that would warrant the effort of developing an expert system for a given task (Figure 3). While some measure of financial return is recognized, the limited availability of an expert resource at every location where a decision must be made is the primary justification for developing expert systems (40:130-131).

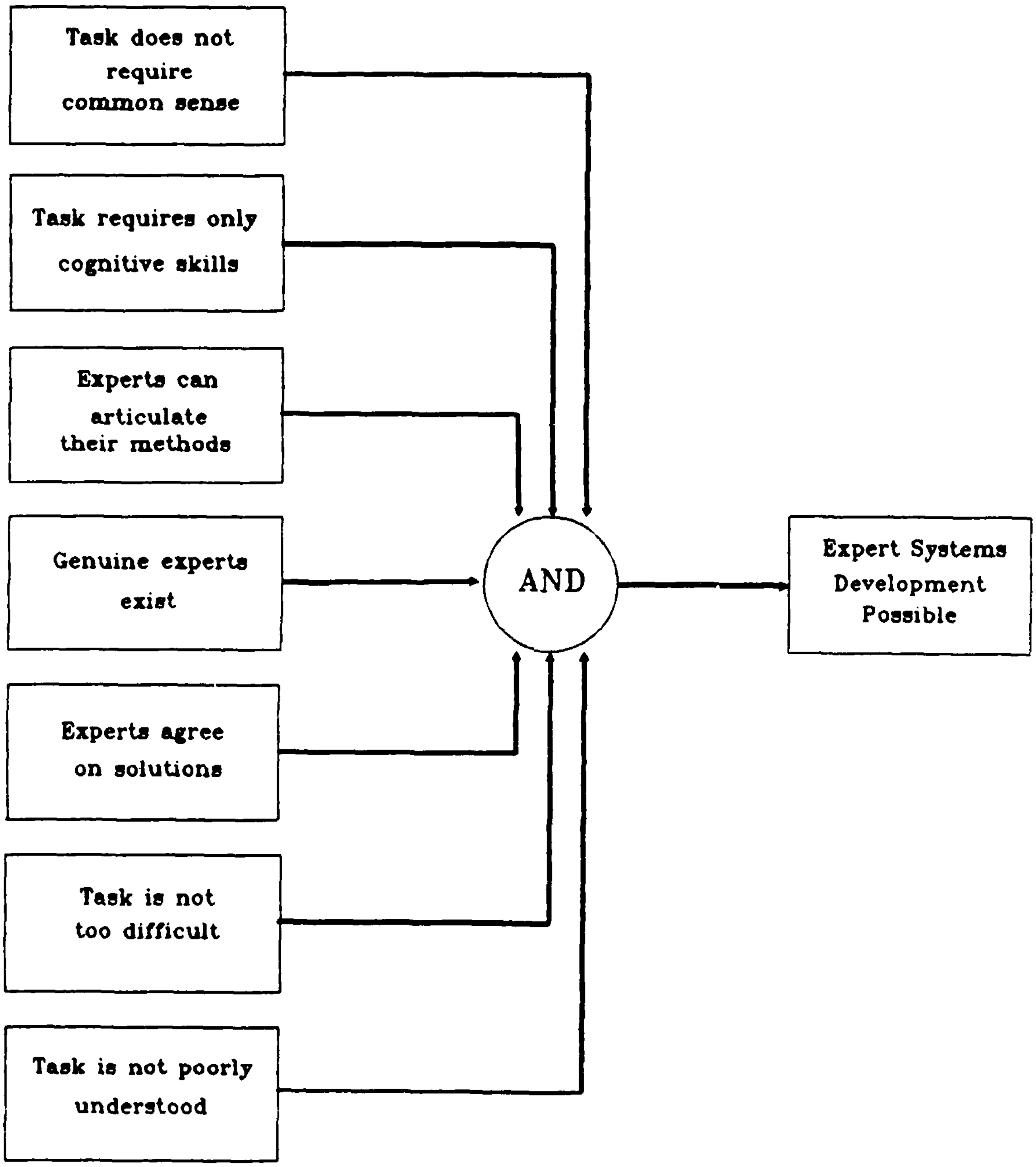

Figure 2. Waterman's Requirements for Expert System Development (Reprinted from 40:129)

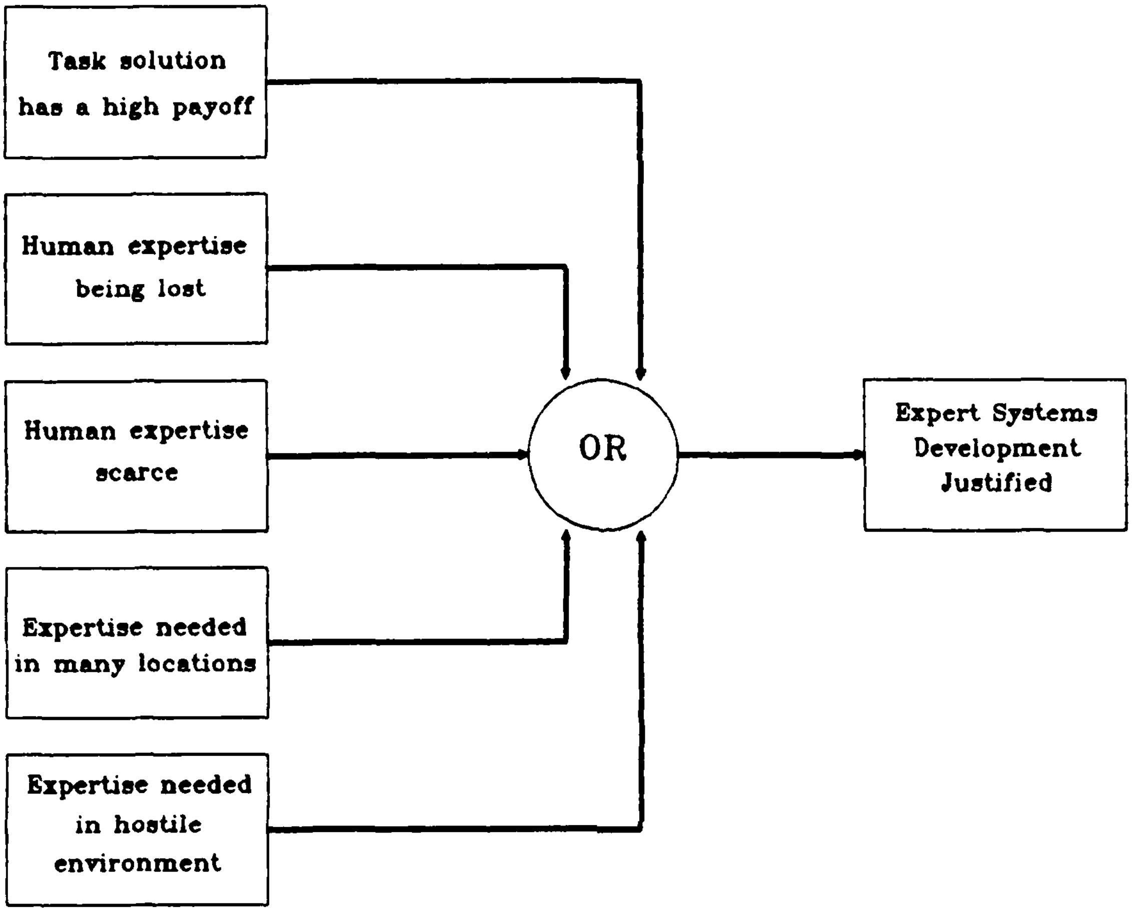

Figure 3. Waterman's Justification for Expert System Development (Reprinted from 40:130)

<u>Appropriate</u>. As with the criteria to determine "possibility," Waterman identifies five characteristics of the task that must exist if an expert system is to be appropriate (Figure 4). The nature of the task refers to the use of symbol manipulation and heuristics versus algorithmic decision processes. An appropriate scope for an expert system project would be one of manageable size, between ten and ten thousand rules, and the project should have some practical value (40:131).

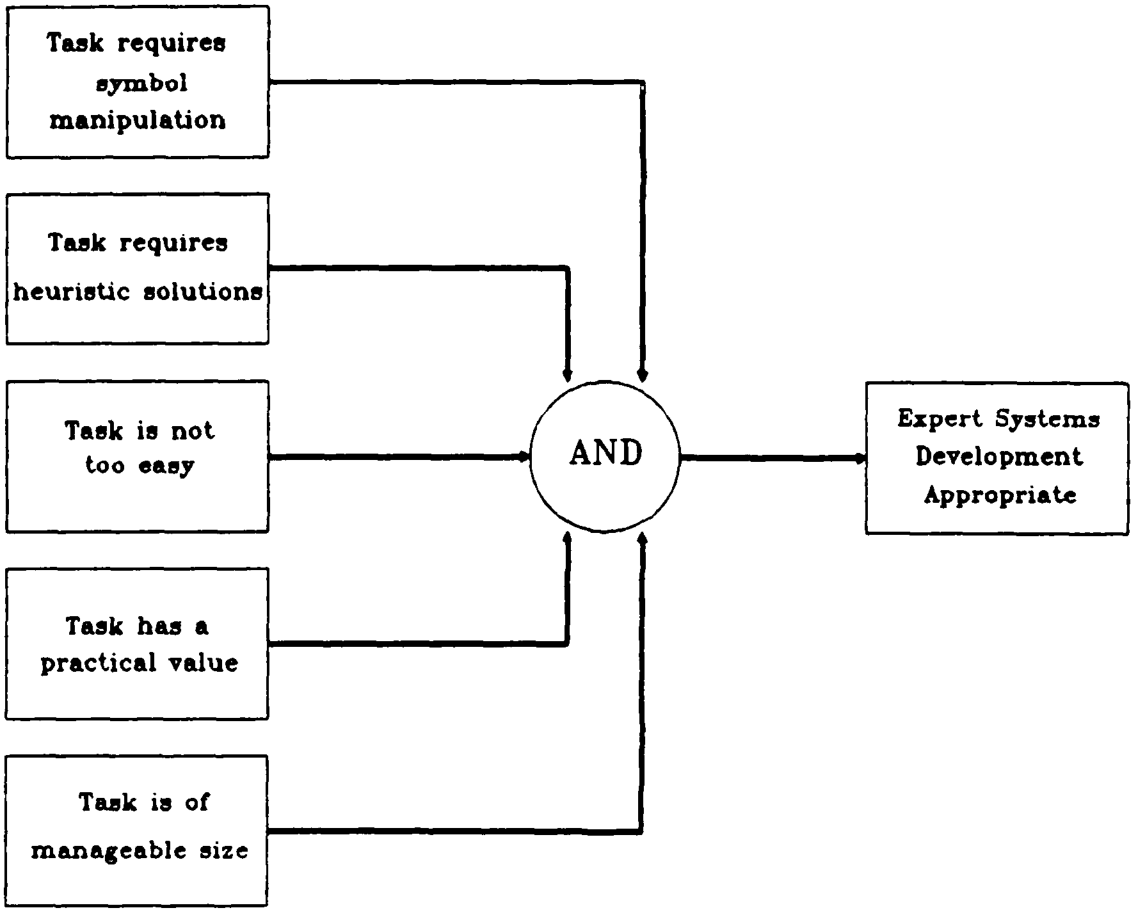

Figure 4. Waterman's Characteristics Appropriate to Expert System Development (Reprinted from 40:132)

Prerau. Perhaps the most thorough list of requirements and desireable characteristics for a problem domain is provided by David S. Prerau in an article he wrote for AI Magazine entitled "Knowledge Acquisition in the Development of Expert Systems" (30:43-51). Prerau discusses requirements for the project, characteristics to identify the best type of problem, traits of potential experts, problem bounds, attributes of domain area personnel, and other desirable features that may help the project end successfully. Many of Prerau's requirements and desirable features parallel on the basic questions proposed by Waterman but contain greater detail.

Schoen and Sykes. Schoen and Sykes feel that many of the existing criteria for determining an appropriate problem domain and task may be overly restrictive, and "less applicable than the existing literature might indicate" (33:106). Schoen and Sykes assert that expert systems are applicable when there is an "existing body of expertise, and this expertise is routinely used for decision-making..." (33:106). In their opinion, if a recognized expert exists, the problem area is narrow and well defined, the performance of an expert is typically superior to that of a novice, and the decision process is "non-algorithmic," the domain is suitable for expert system technology. Table 1 lists the parameters Schoen and Sykes feel define suitable applications for expert system use (33:107).

Table 1
Schoen and Sykes' Key Parameters of Suitable Tasks for Expert System Applications (Reprinted from 33:107)

THE SELECTION PROCESS

Knowledge Content Parameters

- Recognized Expert or Experts
- Bounded Task
- Wide Differences in Performance
- Non-Algorithmic

Summary

In general, a problem domain should involve semi-structured and unstructured decisions. The decision-making processes should be sufficiently difficult to warrant the investment in expert system technology, yet should not be so difficult that it challenges the state of the art for AI. The most important factor in determining an appropriate problem domain is the existence of an expert and that expert's willingness to cooperate in what must be a joint effort to develop an expert system.

IV. Expert System Development

Framework for Development

Quite logically, a planned framework for the development life-cycle of an expert system is recommended. Most development structures found throughout the literature are similar in their general features, yet each possesses unique characteristics that set it apart from the others. All frameworks demonstrate an evolutionary progression from a small prototype to an operational system, similar to the process described by Waterman (Table 2). The following are two different perspectives encompassing three distinct approaches to expert system development.

Harmon and King. Harmon and King identify two approaches to developing expert systems. They choose different approaches for a small and a large expert system.

Small Expert System. Harmon and King somewhat arbitrarily refer to the small expert system as a "knowledge system." They elect to use this distinction because the smaller systems do not necessarily capture the knowledge of human experts. According to Harmon and King, knowledge system knowledge bases may contain other than human "expert" knowledge. However, these authors do not imply that the small knowledge system is to be slighted.

Table 2
Waterman's Description of the Evolution of Expert Systems (Reprinted from 40:140)

Development Stage	Description
Demonstration Prototype	The system solves a portion of the problem undertaken, suggesting that the approach is viable and system development is achievable.
Research Prototype	The system displays credible performance on the entire problem but may be fragile due to incomplete testing and revision.
Field Prototype	The system displays good performance with adequate reliability and has been revised based on extensive testing in the user environment.
Production Model	The system exhibits high quality, reliable, fast, and efficient performance in the user environment.
Commercial System	The system is a production model being used on a regular commercial basis.

> Because small knowledge systems can be created and maintained by the people who actually use them, and because such systems allow individuals with little training to make decisions they could not otherwise make, we think that small knowledge systems will show up in a great many business operations. Moreover, we think their appearance will be welcomed in the same way that managers have welcomed electronic spreadsheet programs. (16:194)

Table 3 summarizes the steps that Harmon and King identify as important to small knowledge system development.

Table 3
Harmon and King's Small Knowledge System Development Steps (Reprinted from 16:178-194)

Step 1. Select a tool and implicitly commit yourself to a particular consultation paradigm.

Step 2. Identify a problem and then analyze the knowledge to be included in the system.

Step 3. Design the system. Initially this involves describing the system on paper. It typically involves making flow diagrams and matrices and drafting a few rules.

Step 4. Develop a prototype of the system using the tool. This involves actually creating the knowledge base and testing it by running a number of consultations.

Step 5. Expand, test, and revise the system until it does what you want it to do.

Step 6. Maintain and update the system as needed.

While this framework is similar to Harmon and King's framework for large expert system development, the role of the expert system developer (often referred to as a knowledge engineer) has been largely replaced in the small system by the expert system development tool and the knowledge of the ultimate system user. Harmon and King feel small systems will improve decision making and enhance productivity in many ways (16:194).

Large Expert System. Harmon and King envision the development of large expert systems as a team effort with special training in knowledge engineering. The expert system development consists of the following phases.

Phase I: Problem selection. Phase I is the foundation upon which the development of the expert system rests. Phase I consists of identifying the problem domain and specific task; identifying a cooperative expert; determining a tentative approach to solving the problem; performing a cost and benefit analysis of potential alternatives; and, constructing a plan to guide the development effort (16:197-201).

Phase II: Prototype development. Phase II is the development of a small scale version or prototype of the desired final expert system. It serves as a validation of important concepts and relationships, and allows the knowledge engineer to become familiar with the problem domain. During Phase II, a development tool is selected and a detailed expert system design is completed (16:201-203).

Phase III: Complete expert system development. Phase III is further development and refinement of the prototype expert system. It consists of enlarging the knowledge base, improving the user interface, reassessing the content of the knowledge base, and monitoring and controlling the expert system's performance (16:203-205).

Phase IV: Expert system evaluation. Phase IV is an evaluation of the expert system against specific criteria identified during the prototyping phase (Phase II) (16:205).

Phase V: Expert system integration. Phase V involves integrating the expert system into the using environment. This encompasses all the efforts necessary to tie the expert system into the present workings of the particular business. This would include integrating the expert system with existing data bases, other available software, and any existing hardware (16:205-206).

Phase VI: Expert system maintenance. This phase allows for any necessary upgrading of the system to meet evolving needs of the using organization. It may include expert system modification or updating of the expert system's knowledge base (16:206-207).

Schoen and Sykes. Schoen and Sykes provide only a single framework for expert system development. Their framework consists of four project development phases, each containing from two to eight steps. Table 4 shows Schoen and Sykes "AI Development Project Phases." Schoen and Sykes emphasize the recursive nature of expert system development. Various steps within each phase are revisited throughout the development life-cycle. Figure 5 illustrates Schoen and Sykes "recursive" expert system development (33:184).

Table 4
Schoen and Sykes' Project Development Life-Cycle Phases of an Expert System (Reprinted from 33:184)

ANALYSIS

1. Application definition
2. Project plan
3. Commitment of resources
4. Initial selection of hardware and software
5. Limited-scope knowledge acquisition
6. Knowledge representation
7. Preliminary system design and coding
8. Demonstration of prototype

DESIGN

1. Refinement of application definition
2. Project plan upgrade
3. Confirmation or modification of hardware and software selection
4. Knowledge acquisition
5. Detailed design
6. Prototype building
7. Evaluation using limited scope examples
8. User and functional inputs
9. Detailed application definition

DECISIONS

1. Use, modify, or reject the prototype
2. Upgrade project plan

VERIFICATION AND DELIVERY

1. Acquire knowledge
2. Complete detailed design
3. Test and validate
4. Train users
5. Field test
6. Release and deliver
7. Maintain the system

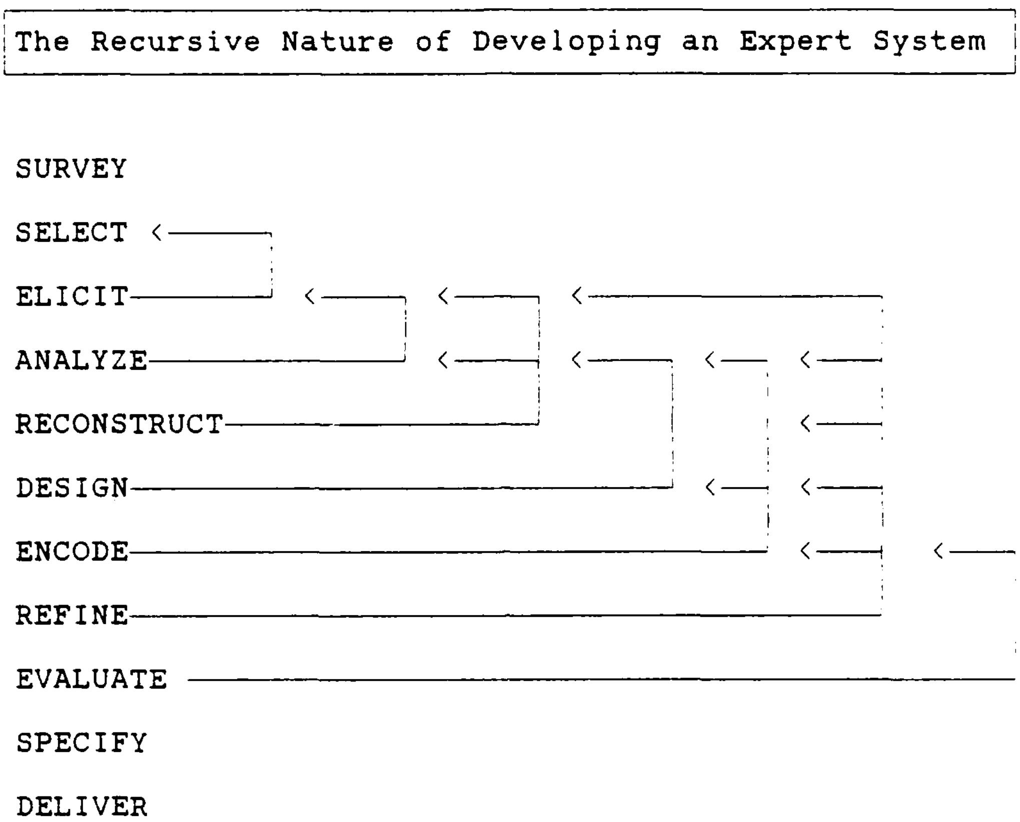

Figure 5. Schoen and Sykes' Recursive Pattern of Expert System Development (Reprinted from 33:104)

<u>Analysis</u>. Schoen and Sykes' analysis portion is similar to Harmon and King's Phases I and II. During this project phase, the problem domain and task area are defined. Appropriate hardware and software are acquired. An initial development plan is adopted. This plan should be sufficiently flexible to allow adaptation to changes in the scope and requirements of the problem domain. Finally, a prototype is constructed to demonstrate the feasibility of the approach (33:186).

Design. The design phase further develops and refines the previous prototype. Additional inputs are obtained from other potential system users. By this point in the development life-cycle, better understanding of the problem should allow refinement of the approach chosen to solve the problem. The output of the design phase should be a usable prototype that will satisfy a "limited-scope problem" (33:186).

Decisions. At this point, the decision is made to proceed with further development of the existing prototype, modify the prototype, or abandon the current approach and try again. The decision is based on inputs from the users and other functional area personnel (33:186).

Verification and Delivery. This is the final stage in development. During this phase, validation and user training should be accomplished. Additionally, some measure of operational testing should be done (33:186-187).

Verification and Validation. Verification and validation are essential steps in the development of expert systems. Verification refers to building the system right, while validation refers to building the right system (28:92).

Verification. Verification involves testing the accuracy of each rule and establishing a justification for each rule in the knowledge base of the expert system (40:160). Verification is accomplished throughout the knowledge acquisition and encoding process. Each rule should be examined for consistency and completeness (26:70). The program should be checked for redundant rules, conflicting rules, unnecessary IF conditions, and circular rules (26:70). Additionally, non-referenced attribute values, illegal attribute values, missing rules, unreachable conclusions, dead-end IF conditions and dead-end goals should be identified and corrected (26:76).

Validation. Validation is a critical part of the overall evaluation process which seeks to assess an expert system's overall value and to answer the question of just how closely the system performs to human expert levels (28:91). Validation should include some form of qualitative and quantitative measure of system performance. Most often the expert system should be validated against human expert performance (28:91).

Summary of the Development Process. As can be seen in the statements by Harmon and King, and Schoen and Sykes, development frameworks involve an iterative, evolutionary approach to expert system development. Development is a process that is marked by familiarization by the knowledge engineer with the problem domain. Familiarization is followed by several knowledge acquisition sessions with the problem domain expert(s). A prototype expert system is then developed, and pending favorable results with the prototype, the knowledge engineer elaborates and embellishes the prototype to create an expert system capable of fulfilling the needs for which it was intended.

Development Tools (Software)

When the decision is made to develop an expert system, a software development tool will be required to help create the system. The two basic types of expert system development tools are programming languages and expert system shells. Each type is discussed separately below.

Programming Languages. Programming languages are used to create new computer software. Expert systems can be created using almost all types of common programming languages, such as BASIC, Fortran, C, Pascal, Forth, and Assembly language (11:824). Most conventional languages are problem oriented and requires a considerable amount of code. They also require considerable design and debugging time (11:824). Expert systems built with conventional languages tend to run slowly because of the complex search and symbol manipulation required (11:824). In contrast to conventional languages, two programming languages have been designed specifically for artificial intelligence applications. They are LISP and PROLOG (38:26). Both have been extensively used to construct expert systems (38:26). LISP and PROLOG are symbolic programming languages that provide operations that manipulate symbolic objects and their interrelations. They also lend themselves to rapid prototyping, a key part of expert system development (11:153).

Expert System Shells. Expert system shells are special software packages created specifically to help build expert systems. They are similar in some respects to conventional software packages like data base management systems or spreadsheets. A shell provides the basic framework in which knowledge can be entered and manipulated in predefined ways (11:824).

Freiling explains that an expert system shell does not contain a knowledge base. The major distinguishing difference between expert systems built using expert system shells is the content of the knowledge base. The inference engine and user interface will work with many different knowledge bases. Knowledge is simply coded in the designated format and entered into the knowledge base (14:45).

Making a Choice. Choosing between a programming language or an expert system shell to develop an expert system depends on a number of considerations. Among these are the characteristics of the problem domain, the operating environment, the experience level of the programmer, time, and money (11:824).

The problem domain characteristics are the primary consideration. They will determine the size of the system, the inference strategy required, and the knowledge representation scheme. For applications with multifaceted dimensions, the expert system may need to be built from scratch. This requires using a programming language to write all components of the expert system, including the inference engine and knowledge representation scheme (38:9).

This can be a complex process which may involve special programmers, special operating environments, extensive hardware, extended development time, and a great deal of money (11:824). Expert system shells, on the other hand, can be very inexpensive and easy to program. Most shells use predetermined representation schemes and may only use one type of inference strategy, a fact which may impose constraints on the methods of organizing knowledge and solving problems (38:9).

Expert Selection

Knowledge acquisition is readily acknowledged to be the most difficult aspect of developing an expert system (23:401). Obviously, the selection of an appropriate expert and the correct use of an appropriate knowledge acquisition technique can have a significant impact on the ultimate success of the expert system (40:128).

Definition of an Expert. An expert is defined as "A person with a high degree of skill in or knowledge of a certain subject. Having or demonstrating impressive skill, dexterity, or knowledge" (25:462). Additionally, Olson feels that experts are further differentiated from the novice in that an expert organizes his knowledge concepts with "...much more depth and many more central associations than novices" (29:152). It is this special knowledge or expertise that the knowledge engineer must somehow acquire and translate into rules to construct the expert system knowledge base.

Expertise. Expertise can be defined as a specialized knowledge and the ability to use that knowledge (5:382). "Expertise is primarily a skill of recognizing, of 'seeing' old patterns in the new problem" (29:152). Olson feels experience is a significant part of this ability to recognize patterns (29:152).

Knowledge. Researchers have attempted to define different forms of knowledge and identify acquisition techniques suitable to the knowledge forms. Generally, acquisition techniques have focused on explicit knowledge. Explicit knowledge is "knowledge of concepts and relations; routine procedures; of facts and heuristic; and classificatory knowledge" (3:144). However, according to Donald Waterman and P.E. Johnson, there exists a phenomenon Johnson identifies as the "paradox of expertise." "The more competent that domain experts become, the less able they are to describe the knowledge they use to solve problems" (40:154). This type of knowledge is referred to as implicit knowledge, knowledge that is used without conscious effort (3:145). Implicit knowledge can be further partitioned into knowledge that at one time was represented explicitly and knowledge that results from a learning or experience base and has never been represented explicitly (3:145).

The Pure Implicit Knowledge Model, developed by Fitts and Anderson and described by Berry, illustrates how expert knowledge can evolve through a three-stage process from a pure explicit form to an implicit form of knowledge:

> Stage I (cognitive stage): The individual learns from instruction and observation.
>
> Stage II (associative stage): The skills learned in Stage I are practiced and refined.
>
> Stage III (autonomous stage): The practiced skills are compiled and used without conscious effort. The experts eventually lose the ability to verbalize their expertise. (3:145)

Schoen and Sykes perceive knowledge as belonging to three general categories: public knowledge, shared knowledge, and private knowledge. This structure divides the explicit-implicit knowledge continuum into three portions. Public knowledge is that knowledge which is readily available. It is usually published, and can be taught. Shared knowledge is the knowledge that develops when a group works together. It does not have to be documented. Schoen and Sykes refer to team sports as an example of a situation that fosters this type of knowledge. Private knowledge is the most personal of the three. It is not easily described by the expert. Certain aspects of this type of knowledge may be referred to as "common sense." This type of knowledge may require different knowledge acquisition techniques to extract (33:96-97).

Expert Selection Criteria. Clearly, one would want to enlist the assistance of the best qualified individual from within a specific problem domain to serve as the source of expertise for the knowledge base. However, the literature suggests other desirable characteristics for the potential knowledge source. An expert should be recognized, and possibly nominated, by his peers for his performance in a particular field. Additionally, an expert should possess the traits of quality performance, quick problem solving ability, specialization, and ability to articulate explanations that one wishes to incorporate in the expert system (18:42). Figure 6 illustrates the desirable characteristics of an expert.

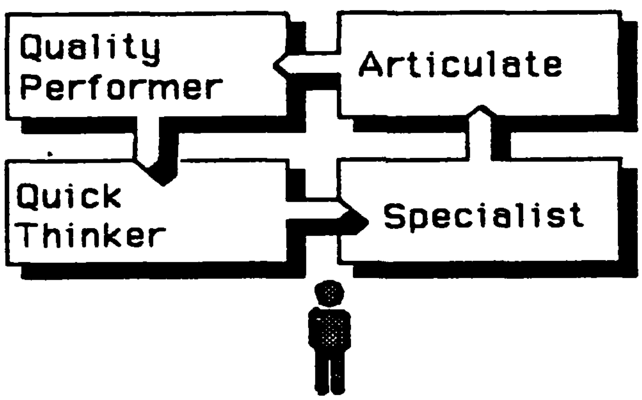

Figure 6. Desirable Characteristics of an Expert (Adapted from 18::42)

The expert should be known for "high-quality" performance. The expert should be able to identify solutions quickly. Specialization and expertise are interrelated, and as such the expert should also be a specialist. Lastly, the expert should be able to explain his actions (18:42).

Knowledge Acquisition Methods

Kim and Courtney define knowledge acquisition as "the process of gathering knowledge about a domain, usually from an expert, and incorporating it into a computer program" (22:269). The power of an expert system comes from the knowledge base (22:269). While this knowledge can come from many sources such as text books, manuals, and data bases, the emphasis in knowledge acquisition has been placed on the human expert (19:53). Because of the time and difficulty associated with extracting knowledge from the human expert, knowledge acquisition has been cited throughout the literature as the critical bottleneck in the development of expert systems (4:228; 3:144; 29:152; 22:269). Since the success of an expert system is dependent on the quality of the knowledge obtained from the expert, the process of obtaining and representing this knowledge is critical (14:158).

Schoen and Sykes also provide a checklist of expert knowledge components that the knowledge engineer should endeavor to identify and define.

1. Symbols and language used
2. Organization and structure of knowledge
3. Elements of knowledge
4. Reasoning methods used
5. Knowledge sources
6. Products or results provided
7. Examples or test cases for use in evaluation (33:110)

Expertise consists of representations accumulated over a lifetime of special experience. With the possible exception of those who teach, few experts spend very much time explaining their knowledge (38:203). To elicit knowledge from the expert, the knowledge engineer must understand the ways in which the expert relates objects, relationships, conditions, constraints, and events within the area of expertise and then apply the appropriate knowledge acquisition tool (38:203).

The literature discusses many methods of knowledge acquisition. These methods can be divided into two categories: direct and indirect (29:153). Direct methods are those which have experts describe the problem solving processes that they can explain directly (29:153). These methods include interviewing, protocol analysis, questionnaires, observation analysis, interruption analysis, drawing closed curves, and inferential flow analysis. Indirect methods rely on the collection of other behaviors that the experts exhibit when solving problems. From these behaviors, the knowledge engineer can make inferences about what the experts must have known to respond the way they did (29:153). Indirect methods include multidimensional scaling, hierarchical clustering, general weighted networks, ordered trees from recall, repertory grid analysis and visual modeling or concept mapping. The following paragraphs discuss each method and lists sources of further information on each method.

Interview. The most common method of knowledge acquisition is the face-to-face interview. Through conversation, experts are asked to verbalize how they solve problems (29:153). Information is collected most often with the aid of a tape recorder, and subsequently transcribed, analyzed, and coded (39:31).

AI researchers have found that the interview is one of the most important tools for facilitating the transfer of human knowledge (29:153; 39:31). Most knowledge engineering sessions begin with an informal interview to get acquainted with the expert and to gain a basic understanding of the basic structure of the problem domain (3:229). Once this has been accomplished, a more structured knowledge acquisition technique is employed.

The greatest advantage of using the interviewing technique is that it is a natural process which is easily understood by both the knowledge engineer and the expert (4:229; 29:153). However, interviewing is often more than just simply sitting down and talking with an expert. The interview relies on the expert's ability to articulate the information used to work through a task. Unfortunately, experts often have a difficult time verbalizing how they go about solving problems (39:31). As a result, the interview is not always a reliable way to obtain complete, objective, or well-organized descriptions of complex cognitive processes.

<u>Sources of Information on Interviews</u>:

Diaper, Dan. <u>Knowledge Elicitation: Principles, Techniques and Applications</u>. New York: Halsted Press, 1989.

Hart, Anna. Knowledge Acquisition for Expert Systems. New York: McGraw-Hill Book Company, 1986.

Waldron, Vincent R. "Interviewing for Knowledge," IEEE Transactions on Professional Communication, 29(2): 31-34, (2 June 1986).

Questionnaire. According to Olson, questionnaires can be a very effective and efficient method of accessing an expert's knowledge. Questionnaires are useful for acquiring explicit knowledge. They can be used to determine objects from the knowledge domain, relationships among those objects, and uncertainties about those relationships (29:154).

Olson further claims that the major advantages of using a questionnaire are that it is less time consuming for the knowledge engineer than the interview, it is an efficient method for gathering information, and the expert can answer the questions at the expert's leisure. The major disadvantage of a questionnaire is that it is unable to pursue unanticipated information (29:154-155).

Sources of Information on Questionnaires:

Sheard, James L. and Brian G. Gnauck. "Questionnaire Design, Administration, and Analysis." Unpublished Report. Air Force Institute of Technology Library, Wright-Patterson AFB, OH.

Graesser, Arther C. and John B. Black. The Psychology of Questions. Hillsdale, New Jersey: Lawerence Erlbaum Associates, 1985.

Task Observation. Task observation involves observing experts work at real problems to determine how they make decisions (29:155). Using the task observation method, the observer will watch the behavior and activities of the experts as they typically proceed through a problem, and then the observer can ask questions (29:155). Recording the experts' performance can be accomplished by simply taking notes, using a tape recorder, or even videotaping the process.

Task observation is a straightforward approach to knowledge acquisition and, like the interview, is easily understood by both the expert and the knowledge engineer. Observations of actual performance can reveal the inference strategy and can also correct misleading or incomplete verbal descriptions of the problem solving process (8:149).

Task observation is not suited to achieving all knowledge acquisition goals, and there are limitations to this method. Access to the people and places to be observed is foremost among the problems (13:22). The problem may also take a considerable amount of time to solve (13:22). Experts may act unnaturally when they are aware that they are being watched (8:148). Additionally, all activities during the observation period are being observed. As a consequence, large quantities of data may be collected from which little actual problem solving knowledge may be useful (8:22).

Sources of Information on Task Observation:

Diaper, Dan. Knowledge Elicitation: Principles, Techniques and Applications. New York: Halsted Press, 1989.

Fraser, Bonnie D. Knowledge Acquisition Methodology. Technical Report. Naval Ocean Systems Center, San Diego, California, June 1987 (AD-A183551).

Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 152-168 (August 1987).

Protocol Analysis. Similar to task observation, protocol analysis involves having an expert perform actual or simulated problem solving scenarios (3:148). Unlike task observation, however, the expert is asked to provide a running verbal commentary on his thought processes as he solves the problem (3:148). One of the knowledge engineer's primary roles is to keep the expert talking, but not to ask specific questions. A detailed analysis of the subsequent transcriptions provides the facts and rules to be used in the knowledge base (22:273).

Protocol analysis can be used to collect both implicit and explicit expert knowledge (29:155; 3:148). The main advantage protocol analysis has over the interview and task observation is that information collected is directly related to the problem solving process, and the knowledge engineer is not required to infer the steps involved in solving the problem (22:273; 29:155).

Protocol analysis is limited to processes that lend themselves to verbalization. Protocol analysis requires a time consuming dissection of the transcripts to produce a usable model of the expert's knowledge (22:273). Other weaknesses of protocol analysis are that the protocols may fail to tap the full range of an expert's knowledge, and the very act of verbalizing the problem solving process may affect the actual way an expert solves a particular problem (3:148).

Sources of Information on Protocol Analysis:

Ericcson, K.A. and H.A. Simon. Protocol Analysis: Verbal Reports as Data. Cambridge, Massachusetts, 1984.

Kim, Jungduck and James F. Courtney. "A Survey of Knowledge Acquisition Techniques and Their Relevance to Managerial Problem Domains," Decision Support Systems 4:269-284 (September 1988).

Olson, Judith R. and Henry H. Rueter. "Extracting expertise from experts: Methods for knowledge acquisition." Expert Systems The International Journal of Knowledge Engineering, 4:152-167 (August 1987).

Interruption Analysis. Interruption analysis is another method for accessing explicit forms of expert knowledge. Like task observation, the expert is observed solving relevant tasks. The expert is not asked to provide a verbal commentary on his decision making process, but whenever the expert does something that the knowledge engineer does not comprehend, the knowledge engineer interrupts the expert and asks exactly what the expert did (29:156).

The main advantage of interruption analysis is that the knowledge engineer is able to capture an expert's knowledge "at the moment the focus of attention" is greatest (29:156). This advantage comes at the expense of interrupting the thought process and risking not being able to restart it. Olson feels that this technique provides the best results when used after a prototype system has been developed, and the system's performance is being compared to that of an expert (29:156).

Source of Information on Interruption Analysis:

Olson, Judith R. and Henry H. Rueter "Extracting expertise from experts: Methods for knowledge acquisition." Expert Systems The International Journal of Knowledge Engineering, 4:152-167 (August 1987).

Drawing Closed Curves. "The method of drawing closed curves is a specialized method for indicating the relationships among those objects that can be assumed to be encoded in a physical space" (29:156). This technique is applicable to any task involving spatial relationships among the objects in a problem domain. Although Olson considers this a direct method of knowledge acquisition, it does not rely on the expert's ability to verbally identify the relationship among objects in a problem domain. Instead, the expert is given several objects to evaluate and is asked to indicate which objects go together. A closed curve is then drawn around those related objects. The closed curves are then compared and evaluated for consistency (29:156-157).

Little more, if any, about this technique is identified in other sources of the literature concerning knowledge acquisition.

Source of Information on Drawing Closed Curves:

Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 152-168 (August 1987).

Inferential Flow Analysis. According to Olson, inferential flow analysis is a direct knowledge acquisition technique to access explicit expert knowledge. Inferential flow analysis can be considered a distinct form of interview. This technique uses specific questions about a knowledge domain to determine cause-and-effect relationships among domain concepts. From the interviews, "causal networks" are developed among the various concepts. This technique is relatively simple to apply and is a powerful tool (29:157).

Source of Information on Inferential Flow Analysis:

Olson, Judith R. and Henry H. Rueter. "Extracting expertise from experts: Methods for knowledge acquisition." Expert Systems The International Journal of Knowledge Engineering, 4:152-167 (August 1987).

Repertory Grid Analysis. According to Olson, repertory grid analysis is one of the most thorough knowledge acquisition techniques available. Repertory grid analysis is classified by Olson as an indirect method capable of drawing upon implicit expert knowledge (29:162). Repertory grid analysis is based on personal construct theory developed by Kelly, and attempts to model human thinking (17:133). "The repertory grid is a representation of the expert's view of a particular problem" (17:134).

Repertory grid analysis consists of some interviewing to identify concepts within the expert's area of expertise. Once concepts are identified, traits that differentiate or group concepts are established, and all the concepts are rated on a scale of from one to three or five, relative to these traits. The extremes of the rating scale represent virtual opposites relative to the trait of interest. Once the rating has been accomplished, the analysts may use some sort of clustering methodology, such as Johnson's hierarchical clustering, to group like concepts. From this, inferences can be made about the relationships of the concepts from the expert's domain of expertise (29:163-164).

The main criticism of this method is its very personal and consequently subjective nature (17:133). This very attribute, however, can be used to advantage. Grids from different experts can be compared and contrasted to gain different perspectives on the same problem (17:133).

Repertory grid analysis is not as complex as it sounds. Interested reader's should consult the sources referenced below to assess the usability of this technique for a given situation.

Sources of Information on Repertory Grid Analysis:

Hart, Anna. Knowledge Acquisition for Expert Systems. New York: McGraw-Hill, 1986.

Olson, Judith R. and Henry H. Rueter. "Extracting expertise from experts: Methods for knowledge acquisition." Expert Systems The International Journal of Knowledge Engineering, 4:152-167 (August 1987).

Concept Mapping. "Concept maps are intended to represent meaningful relationships between concepts in the form of propositions" (27:15). Concept maps are a visual representation of hierarchical relationships among concepts. Concept maps can be used for organizing meanings and illustrating how one perceives relationships. For the purposes of knowledge acquisition, concept mapping is a variation of the interview. Since concept mapping focuses on concepts and relationships among those concepts, it can be use to extract implicit expert knowledge.

As stated, for the purposes of knowledge acquisition, concept mapping is used in conjunction with interviewing techniques to produce a model of an expert's knowledge. Those considerations given to proper interviewing technique should be used when executing the concept mapping technique. However, unlike interviewing, the resulting transcript from a session is in the form of a hierarchical concept map.

Each interviewing session is guided by and builds upon the previous concept map (27:119-133).

As with other indirect knowledge acquisition methods, concept mapping is based on assumption and supporting theories (29:166). "Indirect knowledge acquisition methods can be abused to the extent that their basic assumptions are not met by the data" (29:166). Additionally, Kim feels that this technique does not allow for adequately acquiring the actual reasoning process within the domain of expertise (22:279). Kim also feels that one can never fully capture all of the concepts within an area of expertise, and if one did, the full graphical representation would prove too "unwieldy" to use (22:279).

Source of Information on Concept Mapping:

> Novak, Joseph D. and D. Bob Gowin. Learning How to Learn. New York: Cambridge University Press, 1984.

Multidimensional Scaling. Multidimensional scaling is a technique by which experts judge the similarity of all possible pairs of objects in the problem area. For example, experts might indicate that A is similar to B and that the value of the distance between their similarity is 1. Then the expert may indicate that A is similar to C and the value of that relationship is 3 and so on. This process produces a diagram of relationships among the variables in the problem domain. The diagram can also reveal clustering of variables and outliers. The expert can inspect the diagram and then define the relationships in more detail (29:158-159; 37:1-44).

Multidimensional scaling is an indirect method of knowledge acquisition intended to get at the relationships experts see among the objects they deal with when solving a problem. It is assumed that the distance relationship between these objects is symmetric and that it can be graded with some continuous value. The biggest difficulty with using the technique is that the number of pairwise comparisons that must be made with only a few objects considered is in the hundreds and thousands.

Sources of Information on Multidimensional Scaling:

Johnson, Stephen C. "Hierarchical Clustering Schemes", *Psychometrika*, 32(3): 241-254, (September, 1967).

Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," *Expert Systems - The International Journal of Knowledge Engineering*, 4(3): 152-168 (August 1987).

Shepard, Roger N. and others. *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences*, Volume 1, New York: Seminar Press, 1972.

Hierarchical Clustering. Like multidimensional scaling, hierarchical clustering begins with similarity judgments of the objects in a problem domain. It differ from multidimensional scaling in that the similarity between the objects is not always symmetric and cannot be graded continuously. Hierarchical clustering assumes only that the objects are members of a cluster of objects. Again, like multidimensional scaling, a diagram of the clusters can be created and used by the expert for further evaluation and refinement (20:241-242; 29:159-160).

Hierarchical clustering is another indirect method of knowledge acquisition (29:157). The knowledge it elicits is implicit and is classifactory in nature--that is, experts are able to classify into similar groups those objects they deal with in the problem domain (20:242). One difficulty with this technique, as with multidimensional scaling, is that if the number of objects in the domain is large, the resulting number of similarities can be so enormous that the underlying pattern of the objects in not obvious (20:242).

Sources of Information on Hierarchical Clustering:

Johnson, Stephen C. "Hierarchical Clustering Schemes", Psychometrika, 32(3): 241-254, (September, 1967).

Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 152-168 (August 1987).

Shepard, Roger N. and others. Multidimensional Scaling: Theory and Applications in the Behavioral Sciences, Volume 1, New York: Seminar Press, 1972

General Weighted Networks. Like multidimensional scaling and hierarchical clustering, general weighted networks require that experts provide symmetric distance judgments between pairs of objects in a problem domain. It is assumed that the distance provided by the experts comes from the expert's crossing a single primary path between each pair of objects and possibly a secondary path as well. Two networks are then drawn. The first network, called the minimal connected network, shows the most closely linked objects. The second network, called the minimal elaborated network, shows any additional links among the objects. The two networks are then examined by the experts for any concepts that have a large number of connections to any other objects or that are fully linked into circles (29:160).

This technique is designed to analyze knowledge that can be classified according to the distance between the objects compared. Like the scaling and clustering technique it is difficult to use when the number of objects considered is large (29:160).

Source of Information on General Weighted Networks:

Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 152-168 (August 1987).

Ordered Trees from Recall. According to Olson, Reitman, and Reuter, ordered trees do not deal with distances among objects in a problem area, but they do assume that objects in the domain belong in a cluster. The technique deals with a series of what are called recall trials. Experts are asked to recall object names from a cluster of objects over and over. These recall trials are then examined for similarities and regularities. When objects are recalled together they are referred to as chunks. The chunks are then re-drawn into an ordered tree structure which can be viewed by the expert and revised if necessary (29:161-162; 31:554-579).

This technique is intended to induce the way in which experts organize information in their heads through the systematic inspection of the regularity with which objects are recalled. A major advantage of this technique is that the ordered tree it produces is easier to interpret and more reliable than the representations produced by multidimensional scaling or hierarchical clustering. It is limited to addressing knowledge that can be classified.

Sources of Information on Ordered Trees from Recall:

Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 152-168 (August 1987).

Reitman, Judith S. and Henry H. Reuter. "Organization Revealed by Recall Orders and Confirmed by Pauses," Cognitive Psychology 12: 554-581 (December 1980).

Knowledge Acquisition Problems

There are many good knowledge acquisition techniques that have proven effective at eliciting knowledge for incorporation into expert systems. However, even the best technique may often produce less than perfect knowledge. The reader should be aware of the problems associated with knowledge acquisition.

One of the goals of the knowledge engineer is to access the abstract generalizations of an expert's knowledge

(1:123). However, this process is vulnerable to many problems. One should remember, "Expert judgement is human judgement and as such can be improved" (6:164). Even though the acquisition process involves an "expert" in a given field, even experts make errors.

The process of knowledge acquisition is dependent upon the expert's abilities, availability, and willingness to cooperate (30:44). There are several difficulties associated with those requirements. Human experts may have a difficult time explaining how they make decisions (32: 451). This inability to verbalize how they go about solving a problem is a difficulty in knowledge acquisition cited often in the literature (4:228; 3:144; 29:152; 22:269). The availability of the expert is another reason why knowledge acquisition is difficult (4:228). The process of knowledge acquisition takes time. The expert or the expert's supervisors may not be willing to spend the time required to complete the project (4:228-230).

Unwillingness of an expert to participate in the knowledge engineering project is another potential difficulty in knowledge acquisition. An unwilling expert will cause the knowledge acquisition process to fail (29:153). An expert may be unwilling to participate for any number of reasons. Experts may fear their jobs may be eliminated if a computer can capture their expertise. They may also fear the loss of esteem others hold for them if the task they perform is reduced to something simple through the knowledge acquisition process. Finally, the expert may not know how to explain his or her problem solving expertise and may fear being seen as inarticulate (29:153).

Some obstacles facing the knowledge engineer are unrelated to the experts' abilities to deal with problems in his field of expertise. They occur as the expert explains the various thought processes leading to solutions. These problems are the result of the complexity of knowledge forms, differing frames of reference between the knowledge engineer and the expert, personal biases on the part of each, and the use of leading acquisition techniques (23:401; 1:103).

Corrective actions. According to Cleaves, "Cognitive heuristics are metalevel modes of judgement which may occur outside the awareness of the individual, but nevertheless, influence reasoning and judgement" (6:158). Furthermore, according to Olson, none of the knowledge acquisition techniques identified should be executed without substantial caution being used during the analysis and translation of the expert's knowledge. Olson feels that each type of knowledge acquisition process is susceptible to producing incorrect rules and relations from the expert's knowledge. "The knowledge engineer must make judgments of the suitability of a method for knowledge elicitation to the kinds of knowledge the expert is assumed to possess" (29:167).

Even though the knowledge engineer may encounter problems when developing an expert system knowledge base, there are techniques the knowledge engineer can employ to compensate for the problems. Cleaves identifies two types techniques of corrective measures for the knowledge acquisition process. They are behavioral techniques and mechanical techniques. Behavioral techniques use interviewing and group interaction techniques to facilitate accurate collection of the expert's knowledge. Behavioral techniques focus on the expert. They attempt to create an environment conducive to identifying biases by enhancing the expert's awareness of his thought processes. Group interaction can provide feedback to the individual regarding the acquired information (6:164).

Conversely, mechanical techniques focus on manipulating the data after it has been collected. These techniques allow the expert to change the format of the data to find "a more natural means of expression." Mechanical techniques also allow for the weighing of data collected from different experts based on those experts' experiences and abilities (6:164).

Summary

Direct methods for knowledge acquisition are intended to access the knowledge that the expert can directly articulate. The majority of these methods are time consuming for both the knowledge engineer and the domain expert. Interviewing is reported to be the most prevalent of the direct methods and some form of protocol analysis is also quite common. Indirect methods attempt to get at the knowledge that is difficult for the expert to verbalize. No matter which knowledge acquisition technique is relied upon for the majority of the knowledge acquisition, an informal unstructured interview is recommended for the initial session. This interview can be used to educate the knowledge engineer about the expertise to be acquired, and help to establish a positive relationship between the knowledge engineer and expert. Although there are problems inherent with any technique that may used, it is possible to overcome them.

V. Conclusion

The application of computer technology to virtually any task continues, in part, due to the significant advances in computer software technology. Recent innovations in the field of expert system technology have allowed the application of artificial intelligence technology to problems previously solved only by humans. The growth in applied artificial intelligence, or expert system technology, has taken this previous research technology, and implemented it in those situations historically handled by the human problem-solver. The progress in the field of expert system technology is even allowing the neophyte knowledge engineer to develop expert systems for specific tasks.

The development of an expert system is an involved project requiring patience and determination. Typical development life-cycles involve an iterative process beginning with problem identification, domain expert recruiting, knowledge acquisition, prototype development and, if the process goes well, implementation of an operational system.

While problem definition and expert identification may be the most critical parts to project success, the acquisition of expert knowledge has long been recognized as the choke point in system development. Little guidance is available to help novice knowledge engineers determine which of the many knowledge acquisition techniques may be suitable for their problem domains. In fact, most articles on the subject fail to fully enumerate the many techniques being used. Typically, interviews, task observation, and some form of protocol analysis are the most frequently cited methods. Other literature, however, indicates that many other methods may be used, such as repertory grid analysis, interruption analysis, and multidimensional scaling, to name a few. Granted, these methods may be more complex, and somewhat difficult to employ, but that fact does not imply that they are inappropriate or unusable. It is up to the knowledge engineer to analyze the problem domain, potential experts and their knowledge forms, and expert system development tools before committing to one specific knowledge acquisition technique. It may even take a combination of techniques to effectively capture some types of knowledge.

This paper has attempted to consolidate the widely scattered information about expert system technology and knowledge acquisition techniques. The design, development and implementation of this technology are no longer strictly science fiction, but science applied as one of the latest computer technologies. Expert system technology is a tool that virtually any adept computer user may use to facilitate a specific problem-solving process.

Bibliography

1. Abrett, Glenn. "The KREME Knowledge Editing Environment," International Journal of Man-Machine Studies, 27:103-126 (August 1987).

2. Allen, Mary K. and Omar K. Helferich. Putting Expert Systems to Work in Logistics. Oak Brook, IL: Council of Logistics Management, 1990.

3. Berry, Diane C. "The Problem of Implicit Knowledge," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 144-151 (August 1987).

4. Berry, Diane C. and Donald E. Broadbent. "Expert Systems and the Man-Machine Interface," Expert Systems The International Journal of Knowledge Engineering, 4(3): 228-231 (October 1986).

5. Chignell, Mark H. and James G. Peterson. "Strategic Issues in Knowledge Engineering," Human Factors, 30:381-394 (August 1988).

6. Cleaves, D. A. "Cognitive biases and corrective techniques: proposals for improving elicitation procedures for knowledge-based systems," International Journal of Man-Machine Studies, 27:155-166 (August 1987).

7. Cook, Robert L. "Expert System Use in Logistics Education: An Example and Guidelines for Logistics Educators," Journal of Business Logistics, 10(1): 68-82, (April 1989).

8. Diaper, Dan. Knowledge Elicitation: Principles, Techniques and Applications. New York: Halsted Press, 1989.

9. Duchessi, Peter and others. "Artificial Intelligence and the Management Science Practitioner: Knowledge Enhancements to a Decision Support System for Vehicle Routing," Interfaces, 18: 85-93, (March-April 1988).

10. Eide, Randy D. Knowledge Acquisition for an Expert System in the Air Force Civil Engineering Operations Branch. MS thesis, AFIT/GEM/LSM/88S-5. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1988 (AD-201625).

11. Epp, Helmut and others. "PC Software for Artificial Intelligence Applications," Science, 240: 824-840 (6 May 1988).

12. Ericcson, K.A. and H.A. Simon. Protocol Analysis: Verbal reports as data. Cambridge: Cambridge University Press, 1984.

13. Fraser, Bonnie D. Knowledge Acquisition Methodology. Technical Report. Naval Ocean Systems Center, San Diego, California, June 1987 (AD-A183551).

14. Freiling, Mike and others. "Starting a Knowledge Engineering Project: A Step-by-Step Approach," AI Magazine, 150-164 (Fall 1985).

15. Graesser, Arther C. and John B. Black. The Psychology of Questions. Hillsdale, New Jersey: Lawerence Erlbaum Associates, 1985.

16. Harmon, Paul and David King. Artificial Intelligence in Business Expert Systems. New York: John Wiley & Sons, Inc., 1985.

17. Hart, Anna. Knowledge Acquisition for Expert Systems. New York: McGraw-Hill Book Company, 1986.

18. Hayes-Roth, Frederick; Donald A. Waterman and Douglas B. Lenat. Building Expert Systems. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc. 1983.

19. Hoffman, Robert R. "The Problem of Extracting the Knowledge of Experts from the Perspective of Experimental Psychology," AI Magazine, 53-67 (Summer 1987).

20. Johnson, Stephen C. "Hierarchical Clustering Schemes", Psychometrika, 32(3): 241-254, (September 1967).

21. Keim, Robert T. and Sheila Jacobs. "Expert Systems: the DSS of the Future?" Journal of Systems Management, 37:6-14 (December 1986).

22. Kim, Jungduck and James F. Courtney. "A Survey of Knowledge Acquisition Techniques and their Relevance to Managerial Problem Domains," Decision Support Systems 4: 269-284 (September 1988).

23. Madni, Azad M. "The Role of Human Factors in Expert Systems Design and Acceptance," Human Factors, 30: 395-414 (August 1988).

24. Meister, David. Methods of Eliciting Information from Experts. Technical Report. Naval Personnel Research and Development Center, San Diego, California, October 1987 (AD-A187468).

25. Morris, William. The American Heritage Dictionary of the English Language. Boston: Houghton Mifflin Company, 1980.

26. Nguyen, Tin A. and others. "Knowledge Base Verification," AI Magazine, 8(2): 69-77 (Summer 1987).

27. Novak, Joseph D. and D. Bob Gowin. Learning How to Learn. Cambridge: Cambridge University Press, 1984.

28. O'Keefe, Robert M. and others. "Validating Expert System Performance," IEEE Expert: 81-90 (Winter 1987).

29. Olson, Judith R. and Henry H. Reuter. "Extracting Expertise from Experts: Methods for Knowledge Acquisition," Expert Systems - The International Journal of Knowledge Engineering, 4(3): 152-168 (August 1987).

30. Prerau, David S. "Knowledge Acquisition in the Development of a Large Expert System," AI Magazine, 43-51 (Summer 1987).

31. Reitman, Judith S. and Henry H. Rueter. "Organization revealed by recall orders and confirmed by pauses," Cognitive Psychology, 12:554-581 (December 1980).

32. Saylor, James H. and Patrick H. Waycoff. "Training Over the Horizon," Proceedings of the Society of Logistics Engineers Twenty-Third Annual Symposium, 445-452 (1988).

33. Schoen, Seymour and Wendell G. Sykes. Putting Artificial Intelligence to Work: Evaluating and Implementing Business Applications. New York: John Wiley & Sons, Inc., 1987.

34. Schultheis, Robert A. and Mary Sumner. Management Information Systems: The Manager's View. Boston: Irwin, 1989.

35. Seilheimer, Steven D. "Current State of Decision Support Systems and Expert System Technology," Journal of Systems Management, 39: 14-19 (14 August 1988).

36. Sheard, James L. and Brian G. Gnauck. "Questionnaire Design, Administration, and Analysis." Unpublished report. Air Force Institute of Technology Library, Wright-Patterson AFB OH.

37. Shepard, Roger N. and others. Multidimensional Scaling: Theory and Applications in the Behavioral Sciences, Volume 1. New York: Seminar Press, 1972

38. Teft, Lee. Programming in Turbo Prolog. Englewood Cliffs, New Jersey: Prentice Hall, 1989.

39. Waldron, Vincent R. "Interviewing for Knowledge," IEEE Transactions on Professional Communication, 29(2): 31-34 (2 June 1986).

40. Waterman, Donald A. A Guide to Expert Systems. Reading MA: Addison-Wesley Publishing Company, 1985.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1990

3. REPORT TYPE AND DATES COVERED
Technical Report

4. TITLE AND SUBTITLE
AN INTRODUCTION TO EXPERT SYSTEMS AND KNOWLEDGE ACQUISITION TECHNIQUES

5. FUNDING NUMBERS

6. AUTHOR(S)
James R. Heatherton, Captain, USAF
Todd T. Vikan, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AU-AFIT/LS-TR-90-1

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This report is the by-product of information collected by the authors during research into expert system technology conducted at the Air Force Institute of Technology. That research involved methods for selecting appropriate tools (or "knowledge acquisition techniques") to collect information from experts. In the course of the research, the authores discovered that no single publication discussed all of the collection techniques that a knowledge engineer might want to evaluate. This brief report attempts to remedy that deficiency by consolidating into one document the primary knowledge acquisition techniques used today. For each technique, the authors have provided a short description, evaluation, and bibliography for individuals who want to evaluate a technique in greater depth. The discussion of techniques is introduced by an overview of some issues and architectures of expert system design. It is hoped that this survey will be useful to anyone starting to work with expert systems, as well as to busy managers who want to be certain they have selected the best tool for the important job of knowledge extraction.

14. SUBJECT TERMS
Artificial Intelligence, Expert Systems, Knowledge Acquisition, Knowledge Engineering

15. NUMBER OF PAGES
65

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL